REMARKS

Claims 1-19 were pending in the application. The Examiner has rejected Claims 1-5 and 15-18 under 35 USC 102(e) as anticipated by Hoefelmeyer; has rejected Claims 1, 6-8 and 15-18 as anticipated by van der Made; has rejected Claims 9-11 under 35 USC 103(a) as being unpatentable over van der Made in view of Kolichtchak; has rejected Claims 13, 14 and 19 as unpatentable over Hoefelmeyer in view of Kephart and Lamburt; and has rejected Claim 12 as unpatentable over Hoefelmeyer in view of Kephart. For the reasons set forth below, Applicants believe that the claims, as amended, are patentable over the cited art.

The present invention provides a method, program storage device and apparatus for preventing attacks in data processing systems. The intrusion detection system comprises a host or application based sensor for detecting code based intrusions with a relatively low false-positive rate by monitoring system calls. Malicious code strings related to a detected intrusion are identified, extracted and forwarded to a pattern filter located in the monitored data processing system to prevent further intrusions using said malicious code strings. The malicious code strings may be forwarded to a response server for assembling sets of similar malicious code strings for which signatures are generated to permit identification of all malicious code strings contained in a set. The generated signatures are then distributed to monitored and/or monitoring systems of a protected network to prevent further intrusions using the malicious code strings and variations thereof. The generated signatures of the present invention are not simply the malicious code strings, but are signatures generated for correlated sets of code strings and patterns, as set forth on page 7, lines 9-21.

All of the claims, as amended, expressly recite detecting an intrusion by monitoring system calls. Support for the amendment language is found in the original Specification, for example on page 12, lines 10-15.

The Hoefelmeyer system is directed to a parallel scanning system for detecting malicious code. In order to save time, Hoefelmeyer’s front end processor feeds the incoming communications to multiple scanning computer systems (Col. 4, line 66-Col. 5, line 1), each of which is adapted to scan communications for one or more known malicious code signatures (Col. 2, lines 62-64). When any one of the multiple scanning computer systems recognizes a malicious code signature/string, it generates an alarm.

Hoefelmeyer's system can only recognize known malicious code signatures by comparing incoming signatures/strings to stored lists of known malicious code signatures/strings. Hoefelmeyer cannot detect an intrusion without already knowing the malicious code string. As such, Hoefelmeyer cannot detect intrusions before the intrusions occur. Hoefelmeyer's system does not monitor system calls to identify intrusions. Further, Hoefelmeyer does not teach or suggest identifying a malicious code string after having detected an intrusion by monitoring system calls. Rather, Hoefelmeyer simply recognizes a predetermined code string. Finally, Hoefelmeyer does not extract the malicious code string and forward it to an intrusion limitation system. Rather, Hoefelmeyer's scanning computer system generates an alarm. Hoefelmeyer does teach that a detection management system is configured for creating a "signature" of a piece of malicious code and sending that to a remote detection location (140 of Fig. 1) when the remote location cannot afford to run multiple computer scanning systems. What Hoefelmeyer does is send the malicious code string with identification of the type of intrusion (e.g., Trojan, etc) for the remote site to use in its monitoring/pattern matching. Hoefelmeyer is not teaching or suggesting that a signature other than the detected malicious code signature be generated.

Anticipation under 35 USC 102 is established only when a single prior art reference discloses each and every element of a claimed invention. See: In re Schreiber, 128 F. 3d 1473, 1477, 44 USPQ2d 1429, 1431 (Fed. Cir. 1997); In re Paulsen, 30 F. 3d 1475, 1478-1479, 31 USPQ2d 1671, 1673 (Fed. Cir. 1994); In re Spada, 911 F. 2d 705, 708, 15 USPQ2d 1655, 1657 (Fed. Cir. 1990) and RCA Corp. v. Applied Digital Data Sys., Inc., 730 F. 2d 1440, 1444, 221 USPQ 385, 388 (Fed. Cir. 1984). Since the Hoefelmeyer patent does not teach the claimed steps and means for monitoring system calls to detect intrusions, followed by identifying the malicious code string related to a detected intrusion, extracting the malicious code string and forwarding the malicious code string to an intrusion limitation system, it cannot be concluded that Hoefelmeyer anticipates the invention as claimed.

The van der Made patent is also directed to detecting malicious code. However, what van der Made provides is a virtual machine which simulates the functionality of the central processing unit and executes received code to determine if the received code is malicious. The virtual machine runs the code, generates a behavior pattern of the executed received code, and analyzes the behavior patterns to determine if they are characteristic of an intrusion. The van der Made method fully executes the code in a simulation environment, thereby protecting the CPU. van der Made does not detect intrusion by monitoring system calls to identify the intrusive code before it executes; rather, van der Made fully executes the intrusive code and identifies the code as malicious once it sees the full behavior pattern. Further, once van der Made detects a behavior pattern that is associated with malicious code, based on comparison to known behavior patterns of known malicious code, the virtual machine is shut down and its memory resources are released (Col. 9, lines 25-33). Accordingly, the code cannot execute in the actual computing system. van der Made does not teach or suggest extracting the malicious code string and forwarding the malicious code string to an intrusion limitation system.

Applicants reiterate that anticipation can only be established if one reference teaches each and every claim feature. Since the van der Made patent does not teach the claimed steps and means for monitoring system calls to detect intrusions, followed by identifying the malicious code string related to a detected intrusion, extracting the malicious code string and forwarding the malicious code string to an intrusion limitation system, it cannot be concluded that van der Made anticipates the invention as claimed.

With regard to the rejections based on a combination of teachings, Applicants note that all of the 103 obviousness rejections cite either van der Made of Hoefelmeyer as the primary reference. Applicants rely on the foregoing analysis of those patents and maintain that neither van der Made nor Hoefelmeyer teaches or suggests the features of the independent claims. Further, the additionally cited references do not teach those features which are missing from van der Made and Hoefelmeyer.

The Examiner has cited paragraph [0032] of Kolichtchak as teaching inspecting a stack as claimed in Claims 9-11. The cited paragraph, however, teaches the following:

> [0032] If the fault address is the execution address, the process is most likely malicious code, and the method 400 logs (440) and/or terminates the program creating that code. In some embodiments, only the logging step 440 is performed, and the method 400 returns (455) immediately after the logging step 440. In other embodiments, the attempted buffer overflow attack is both logged (440) and terminated. More specifically, the termination process may involve injecting (445) termination code in the current process and changing (450) the return address. In still other embodiments, the method 400 may skip the logging step 440 and simply

> terminate the process without logging. Optionally, the termination process may involve prompting a human operator whether to proceed with the termination.

Applicants respectfully assert that the cited passage from Kolichtchak does not teach inspecting a stack to retrieve an address leading to a malicious code string. Rather, the passage teaches logging and terminating code, possibly with a change of return address. Such is not the same as or suggestive of inspecting a stack to retrieve the address, let alone of locating multiple locations on the stack (Claim 10) and scanning from opposite directions to extract a malicious code string (Claim 11).

With respect to the citation of the Kephart patent, Applicants respectfully assert that the Kephart patent does not qualify as prior art against the present application. The pending rejections of Claims 12, 13, 14 and 19 use Hoefelmeyer in view of Kephart to deny patentability of the present invention under 35 USC 103, because of an assumption that Kephart would qualify as prior art under 35 USC 102. However, this assumption is incorrect. As inspection of Kephart reveals that the Kephart patent is currently assigned to International Business Machines Corporation (IBM). The subject matter of Kephart was owned by IBM and subject to an obligation of assignment to IBM at the time that the present invention was made. In addition, at the same time, the present invention was owned by IBM and subject to an obligation of assignment to IBM. Hence, the present invention and the subject matter of Kephart were owned by a common assignee (i.e., IBM) at the time that the present invention was made.

Claims 13, 14 and 19 are rejected as unpatentable over Hoefelmeyer in view of Kephart and further in view of the Lamburt patent. The Lamburt patent is cited for its teachings regarding the use of edit distance algorithms. Applicants respectfully assert that the addition of edit distance algorithms to Hoefelmeyer (with the Kephart patent teachings being unavailable as prior art against the pending claims) would not result in the invention as claimed. As detailed above, Hoefelmeyer does not teach monitoring system calls to detect intrusions, followed by identifying the malicious code string related to a detected intrusion, extracting the malicious code string and forwarding the malicious code string to an intrusion limitation system. Moreover, one having skill in the art would not be motivated to modify the Hoefelmeyer patent system with Lamburt's edit distance teachings, since Hoefelmeyer is not identifying and comparing multiple code strings. Each of Hoefelmeyer's scanning systems simply recognizes a string and generates an alarm. There are no teachings or suggestions that multiple strings be recognized and their similarities considered.

For a determination of obviousness, the prior art must teach or suggest all of the claim limitations. "All words in a claim must be considered in judging the patentability of that claim against the prior art" (*In re Wilson*, 424 F. 2d 1382, 1385, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970). If the cited references fail to teach each and every one of the claim limitations, a *prima facie* case of obviousness has not been established by the Examiner. Since none of the cited references teaches the claimed steps and means for monitoring system calls to detect intrusions, followed by identifying the malicious code string related to a detected intrusion, extracting the malicious code string and forwarding the malicious code string to an intrusion limitation system, obviousness has not been established. Moreover, Applicants respectfully assert that the Kolichtchak patent publication does not contain the teachings that the Examiner cites it for; that the Kephart patent is not available as a prior art reference; and, that one having skill in the art would not look to the Lamburt patent teachings to modify Hoefelmeyer. Clearly, obviousness cannot be maintained.

Based on the foregoing amendments and remarks, Applicants respectfully request entry of the amendments, reconsideration of the rejections, and issuance of the claims.

Respectfully submitted,

Swimmer, et al

By: */Anne Vachon Dougherty/*
Anne Vachon Dougherty
Reg. No. 30,374
Tel. (914) 962-5910